

PAT-NO: JP402238558A
DOCUMENT-IDENTIFIER: JP 02238558 A
TITLE: BOOT SYSTEM FOR PARALLEL COMPUTER
PUBN-DATE: September 20, 1990

INVENTOR-INFORMATION:

NAME
SHIMIZU, TOSHIYUKI
ISHIHATA, HIROAKI

ASSIGNEE-INFORMATION:

NAME	COUNTRY
FUJITSU LTD	N/A

APPL-NO: JP01060060

APPL-DATE: March 13, 1989

INT-CL (IPC): G06F015/16, G06F009/445 , G06F013/00

ABSTRACT:

PURPOSE: To increase the number of processor elements by executing successively the boot programs received from a communication port via a CPU at the side of each processor element after a host computer detects that all processor elements has accesses to the communication port.

CONSTITUTION: In a reset state an address decoder 4c decodes the address data received from a CPU 4a and has an access to a communication port 4b. A host computer 1 sends a boot program to each processor element PE 4 when the computer 1 detects that all processors PE 4 has accesses to the port 4b. Then

each PE 4 carries out successively the boot programs received from the port 4b via the CPU 4a. Thus the computer 1 gives a program to each PE 4 in an initial program load IPL state. As a result, each PE requires no ROM storing an IPL program and the number of elements PE is increased.

COPYRIGHT: (C)1990,JPO&Japio

⑫ 公開特許公報 (A) 平2-238558

⑬ Int. Cl.⁵
 G 06 F 15/16 9/445 13/00 識別記号 420 S 305 A 庁内整理番号 6745-5B 8840-5B 7361-5B G 06 F 9/06 420 J
 ⑭ 公開 平成2年(1990)9月20日
 審査請求 未請求 請求項の数 1 (全10頁)

⑮ 発明の名称 並列計算機のブート方式

⑯ 特願 平1-60060
 ⑰ 出願 平1(1989)3月13日

⑱ 発明者 清水 俊幸 神奈川県川崎市中原区上小田中1015番地 富士通株式会社
内

⑲ 発明者 石畠 宏明 神奈川県川崎市中原区上小田中1015番地 富士通株式会社
内

⑳ 出願人 富士通株式会社 神奈川県川崎市中原区上小田中1015番地

㉑ 代理人 弁理士 井島 藤治 外1名

月日 系列番号

1. 発明の名称

並列計算機のブート方式

2. 特許請求の範囲

1個のホスト計算機(1)と複数個のプロセッサエレメント(4)とがバス(3)を介して接続された分散メモリ型並列計算機において、各プロセッサエレメント(4)内に、
CPU(4a)と、
バス(3)と接続された通信ポート(4b)と、
プロセッサエレメントのリセット時にCPU(4a)から出力されるアドレスをデコードして通信ポート(4b)をアクセスするアドレスデコーダ(4c)とを備えし、

全てのプロセッサエレメント(4)が通信ポート(4b)をアクセスしたことをホスト計算機(1)側で検知したら、ホスト計算機(1)から各プロセッサエレメント(4)に対してブートプログラムを送出し、

各プロセッサエレメント(4)側では、通信ポート

ト(4b)から入力されるブートプログラムをCPU(4a)により順次実行するように構成したことを持てとする並列計算機のブート方式。

3. 発明の詳細な説明

〔概要〕

1個のホスト計算機と複数個のプロセッサエレメントとがバスを介して接続された分散メモリ型並列計算機のブート方式に関し、

分散メモリ型並列計算機のプロセッサエレメントの数を増やすという利点を十分に生かすことができるようになりますことを目的とし、

各プロセッサエレメント内に、CPUと、バスと接続された通信ポートと、プロセッサエレメントのリセット時にCPUから出力されるアドレスをデコードして通信ポートをアクセスするアドレスデコーダとを備えし、全てのプロセッサエレメントが通信ポートをアクセスしたことをホスト計算機側で検知したら、ホスト計算機から各プロセッサエレメントに対してブートプログラムを送出し、各プロセッサエレメント側では、通信ポート

から入力されるブートプログラムをCPUにより順次実行するように構成する。

【産業上の利用分野】

本発明は1個のホスト計算機と複数個のプロセッサエレメントとがバスを介して接続された分散メモリ型並列計算機のブート方式に関する。

近年、コンピュータシステムの高速化が要求されている。高速化の一つの実現法として、並列計算機が用いられる。ここで、並列計算機とは、プログラムを実行する計算要素（プロセッサエレメント； Processor Element、以下略してPEと記す）を複数個結合して一つの計算機を構成したものである。この種の並列計算機には、大きく分けて2つの実現方法が考えられる。一つは複数のPEで大きなメモリを共有する共有メモリ型並列計算機であり、もう一つはPE毎に独立したメモリをもつ分散メモリ型並列計算機である。後者の分散メモリ型並列計算機は、PEの数を大きくすることが可能であるという特徴をも

っている。ところが、一つのPEのハード盤が大きくなるとこの特徴を生かすことができなくなる。このため、PEのハードウェア盤は小さく抑える必要がある。また、PEの数が増加するに伴い、効率的なブート方法が要求されている。

【従来の技術】

第7図は、従来の分散メモリ型並列計算機の構成ブロック図である。1個のホスト計算機1と複数個のPE2とがバス3を介して接続されている。第8図は各PEの内部構成例（従来）を示す図である。PEは、図に示すようにバス3と接続された通信ポート2a, RAM2b, CPU2c, ROM2d及びこれらを接続する内部バス2eより構成されている。ROM2d内にはブートアップ（IPL：イニシャル・プログラムロード）用のプログラムが格納されている。

このように構成された分散メモリ型並列計算機のブートアップ（IPL）は、PE内に用意されたROM2dに格納されたプログラムによって行

- 3 -

- 4 -

われる。第9図は、従来のブートアップの手順を示すフローチャートである。先ず、ユーザがホスト計算機1を初期化する（S1）。その後、ホスト計算機1はIPLを開始する（S2）。次に、ホスト計算機1はIPLの一つの手順としてPE2を初期化する（S3）。

各PE2はROM2d内に格納されているIPLプログラムの実行を開始する（S4）。次に、ホスト計算機1はPE2に対してOS等をバス3を介して送信し、各PE2はIPLの実行の過程で、OS等をホスト計算機1からバス3を介して受信する（S5）。そして、各PE2はIPLを終了し、動作を開始し、ホスト計算機1はIPLを終了し、動作を開始する（S6）。ここで、動作とは本来の並列処理動作をいう。

【発明が解決しようとする課題】

従来の方式では、各PE毎にブートプログラムを書き込んだROMを用意し、そのプログラムによりブートアップ（IPL）を行っていた。しかし

ながら、この構成をとるとROMの周辺等を含め、ある程度のハードウェアが必要となる。また、PEの数だけROM等を用意せねばならず、システム作製時のコストアップ及び動作時の信頼性の低下にもつながる可能性がある。以上により、従来システムではPEの台数を増やせるという利点を十分に生かすことができなかった。

本発明はこのような課題に鑑みてなされたものであって、分散メモリ型並列計算機のPEの数を増やせるという利点を十分に生かすことができる並列計算機のブート方式を提供することを目的としている。

【課題を解決するための手段】

第1図は本発明方式の原理ブロック図である。第7図と同一のものは、同一の符号を付して示す。図において、1はホスト計算機、3はバス、4はバスに接続された複数個のPEである。各PE4内には、CPU4aと、バス3と接続された通信ポート4bと、PEのリセット時にCPU4aか

- 5 -

—562—

- 6 -

ら出力されるアドレスをデコードして通信ポート4 bをアクセスするアドレスデコーダ4 cと、RAM 4 dより構成されている。図では1つのPEについてその内部構成を示しているが、他のPEについても同様である。

【作用】

リセット時には、アドレスデコーダ4 cはCPU 4 aから出力されるアドレスデータをデコードして通信ポート4 bをアクセスするようとする。そして、全てのPE 4が通信ポート4 bにアクセスしたことをホスト計算機1側で検知したら、ホスト計算機1から各PE 4に対してブートプログラムを送出し、各PE 4側では、通信ポート4 bから入力されるブートプログラムをCPU 4 aにより順次実行するようとする。このような構成とすることにより、IPL時のプログラムは各PE 4に対してホスト計算機1から提供されるので、各PE 4内にIPLプログラムを格納したROMが不要となる。従って、本発明方式によれば分散

メモリ型並列計算機のPEの数を増やせるという利点を十分に生かすことができる。

【実施例】

以下、図面を参照して本発明の実施例を従来例と対比しつつ詳細に説明する。

本発明はPEのCPUから見えるアドレス空間のデコードの方法を工夫することにより、ROMを必要としない方式としたものである。第2図はアドレス空間を示す図であり、(イ)は従来のアドレス空間を、(ロ)は本発明によるアドレス空間をそれぞれ示している。ここでは、次の仮定をしている。まずPEのCPUはホスト計算機から初期化(リセット)されると、アドレス0000#(#=16進を示す)から命令を取り出し、実行を開始する。バスからのデータは、ポート(アドレスF000#)を読むことにより受取る。アドレスは全て16進であり、図に示す値は例示である。

従来のアドレス空間は、(イ)に示すようにアド

- 7 -

レス0000#～2000#はROMに割当てられており、ここにIPLプログラムが格納されていた。後の2000#～F000#まではRAM領域とPORT(ポート)領域が適宜割当てられていた。そして、PEが初期化されると、PE内のCPUはアドレス0000#から2000#の間に置かれたROMに格納されているIPLによって動作に必要なOS等(これらはホスト計算機により作成される)をポートから読み出し、RAMに書きしていく。

これに対し、本発明の場合には(ロ)に示すようにアドレス0000#から2000#までをポートのアドレスとしている。従って、PEが初期化されてPE内のCPUが0000#からアドレスを出力すると、第1図に示したアドレスデコーダがこのアドレスをデコードしてポートアドレスに変換し、通信ポートをアクセスするようとする。この間に、ホスト計算機からIPLのプログラムを各PEに対して送出し、各PEでは通信ポートを経由してCPUにそのプログラムを与え、IPL

- 8 -

Lを実行させるのである。従って、本発明によればIPLプログラムを格納したROMは必要ないことになる。

次に、ホスト計算機がPEに送るデータとPEのCPUが実行する命令の関係を更に詳細に説明する。ここでは、PEのCPUが実行する命令を以下のように定義する。

S T A D D R ; アドレス(ADDR)にレジスタの値を書き込む

L D A D D R ; アドレスの(ADDR)の値をレジスタに読み込む

また、ホスト計算機がPEに送るOSのデータ列をOS0, OS1, … OSZ(OSZが最終データ、データの個数は100#と仮定)と書き表すものとすると、従来方式によりホスト計算機がPEに送るデータとPEのCPUが実行する命令列は、第3図に示すようなものとなる。時刻t1からt2までの間がホスト計算機から各PEにOSを送信しているシーケンスである。

第4図はホスト計算機がPEに送るデータとP

- 9 -

- 10 -

EのCPUが実行する命令例（本発明）を示す図である。従来例では、第3図に示すようにホスト計算機からはOS命令のみが与えられているだけであったが、第4図の本発明の場合にはホスト計算機はOS命令のみならずLD F000。なる命令とST 2000。なる命令を送っている。これら命令は、従来方式では内蔵のROMから与えられていたものである。PE側ではこのLD命令が通信ポートから受取られるとCPUの命令として実行される。つまり、PE側ではCPUから出力されるアドレス0000。～2000。を全て通信ポートのアドレスF000。に変換し、通信ポートから入力されるデータをCPUが実行すべき命令として取り込み実行を進めていくものである。

第4図において、時刻t₁からt₂の範囲がホスト計算機から各PEにOSを送信しているシーケンスである。前述したように、PEのCPUが実行する命令もホスト計算機からPEに送り出されている。このことは言い換えれば、従来ROM

- 11 -

を付して示す。図では、PEを1個しか示していないが、実際にはバス3に複数個接続されている。ホスト計算機1は、CPU1a、メモリ1b及びバス3を介してPE4との接続制御を行なうインターフェイス部1cより構成されている。このインターフェイス部1cには、バス3が接続される他に制御線5が接続されている。この制御線5は各PE4とも接続されている。PE4において、4eはバス3を介してホスト計算機1との接続制御を行なうインターフェイス部、4fはPE4内の内部バスである。第1図で示した通信ポート4bはインターフェイス部4eに含まれる。このよう構成されたシステムの動作を説明すれば、以下のとおりである。

第6図は本発明によるブートシーケンスを示す図である。以下、このシーケンス図に沿って第5図に示すシステムの動作を説明する。先ず、インターフェイス部1cを介してホスト計算機1からPEへのリセット信号が出力される（①）。一方、PE4側では、インターフェイス部4eを介して

に格納していた IPLをホスト計算機から送り出した命令で行っていることになる。従って、各PE内にROMを用意する必要がなくなったのである。このことが可能となったのは、前述したアドレスのデコードの工夫がポイントとである。更に説明する。PE内のCPUは、初期化後アドレス0000。から実行を始める。

そして、アドレス0000。から命令を読み、実行し、次にはアドレス0001。から命令を読み実行する。このように、アドレスを1つずつ更新しながら実行する。従来は、アドレス0000。からのアドレス空間にIPLを書込んだROMを置くことにより、IPLを実行していた。本発明では、この空間に通信ポートを割当て、CPUが初期化後、命令をアドレス0000。から読みもうとすると、通信ポートのデータ、即ちホスト計算機からバスを介して送られてくる命令が読み込まれることになる。

第5図は本発明の一実施例を示す構成ブロック図である。第1図と同一のものには、同一の符号

- 12 -

送られたきたリセット信号を受けて内部の状態をリセットして初期化する（（1））。リセットされると、CPU4aはアドレス0000。から命令をフェッチして実行するようになっている。そこで、CPU4aはアドレス0000。をアドレスデータとして出力する。このアドレスデータはアドレスデコーダ4cによってインターフェイス部4e内の通信ポートをアクセスする信号に変換される。この結果、通信ポートがアクセスされる（（2））。しかしながら、制御線5を介してACK信号（確認信号）がまだ有効になっていないのでそのままホールド状態となる（（3））。

一方、ホスト計算機側では、全てのPE4が通信ポート4bをアクセスするのをインターフェイス部1cを介してCPU1aにより監視している。そして、全てのPEが通信ポートをアクセスするのを待ってPEの第1命令を通信ポート4bに出力する（②）。また、それと同時に制御線5のACK信号を有効にする（③）。

PE側ではACK信号が有効になるまでホール

- 13 -

- 14 -

ドされていたが、ACK信号が有効になったのを受けてCPU4aが第1命令を読み実行する((4))。次に、CPU4aが第2の命令をフェッチするためのアドレス信号0001を出力すると、このデータは再度アドレスデコーダ4cにより通信ポート4bをアクセスする信号に変換され、通信ポート4cをアクセスする((5))。この時、ACK信号は無効状態になっているので、ACK信号が有効になるまでホールドされる((6))。

ホスト計算機側では、全てのPEが通信ポートをアクセスするのを待って、PEの次の命令を通信ポート4bに出力する((4))。それと同時に、制御線5のACK信号を有効にする((5))。

PE側では、第2命令を通信ポート4bを介して読み実行する((7))。このようにしてPE側では、CPU4aがフェッチする命令(命令アドレス)がFFFを越えない間、(5)、(6)、(7)を繰り返す((8))。一方、PE側ではブートシーケンスを終了するまで④、⑤

を繰り返す((6))。OSをホスト計算機から送る場合には、第4図で示したように、ホスト計算機は、PEが実行すべき命令に合わせて、OSのデータを送ればよい。

【発明の効果】

以上、詳細に説明したように、本発明によればPEがリセットされてからPE内のCPUが命令フェッチ用に出力するアドレスをデコードして通信ポートをアクセスする信号に変換してやり、IPLのための命令を通信ポート経由でホスト計算機から貰って実行する構成とすることにより、PE内のROMを不要とすることができる。従って、本発明によれば分散メモリ型並列計算機のPEの数を増やせるという利点を十分に生かすことができるようになる。

4. 図面の簡単な説明

第1図は本発明方式の原理ブロック図、

第2図はアドレス空間を示す図、

第3図はホスト計算機がPEに送るデータとP

- 15 -

- 16 -

EのCPUが実行する命令例(従来)を示す図、

第4図はホスト計算機がPEに送るデータとP

EのCPUが実行する命令例(本発明)を示す図、

第5図は本発明の一実施例を示す構成ブロック図、

第6図は本発明によるブートシーケンスを示す図、

第7図は従来の分散型並列計算機の構成ブロック図、

第8図は各PEの内部構成例(従来)を示す図、

第9図は従来のブートアップの手順を示すフローチャートである。

第1図において、

1はホスト計算機、

3はバス、

4はPE、

4aはCPU、

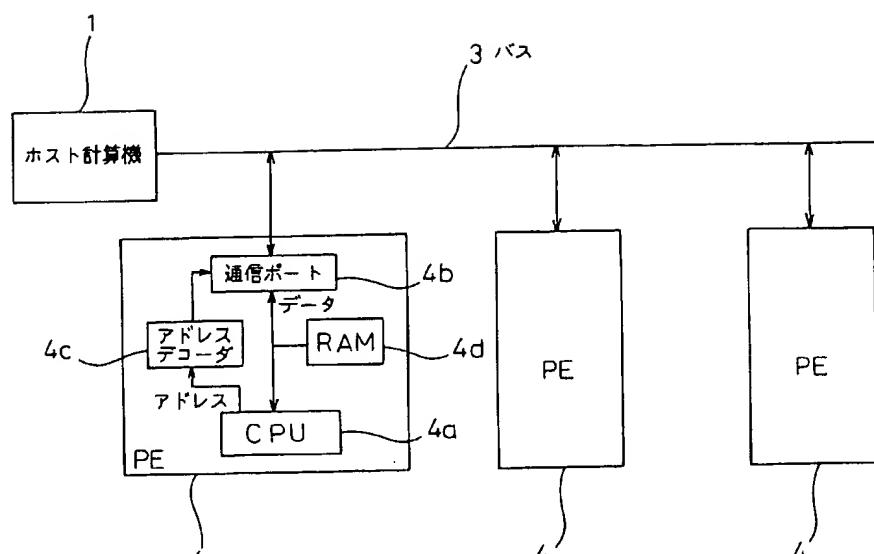
4bは通信ポート、

4cはアドレスデコーダ、

4dはRAMである。

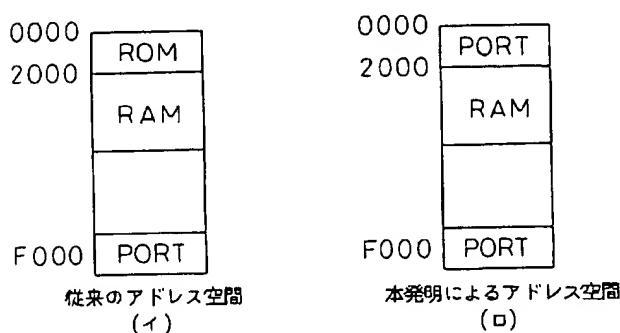
- 17 -

-565-



本発明方式の原理ブロック図

第1 図



アドレス空間を示す図

第2 図

ホスト計算機がPEに送るデータ	PEのCPUが実行する命令
t1 時間 OS0	LD F000 ; OS0がレジスタに入る ST 2000 ; OS0を2000番地に書く
OS1	LD F000 ; OS1がレジスタに入る ST 2001 ; OS1を2001番地に書く
OS2	LD F000 ; OS2がレジスタに入る ST 2002 ; OS2を2002番地に書く
OS3	LD F000 ; OS3がレジスタに入る ST 2003 ; OS3を2003番地に書く
OS4	⋮
⋮	⋮
OSZ	LD F000 ; OSZがレジスタに入る ST 20FF ; OSZを20FF番地に書く ⋮ ; OSのスタートアドレスから実行を開始

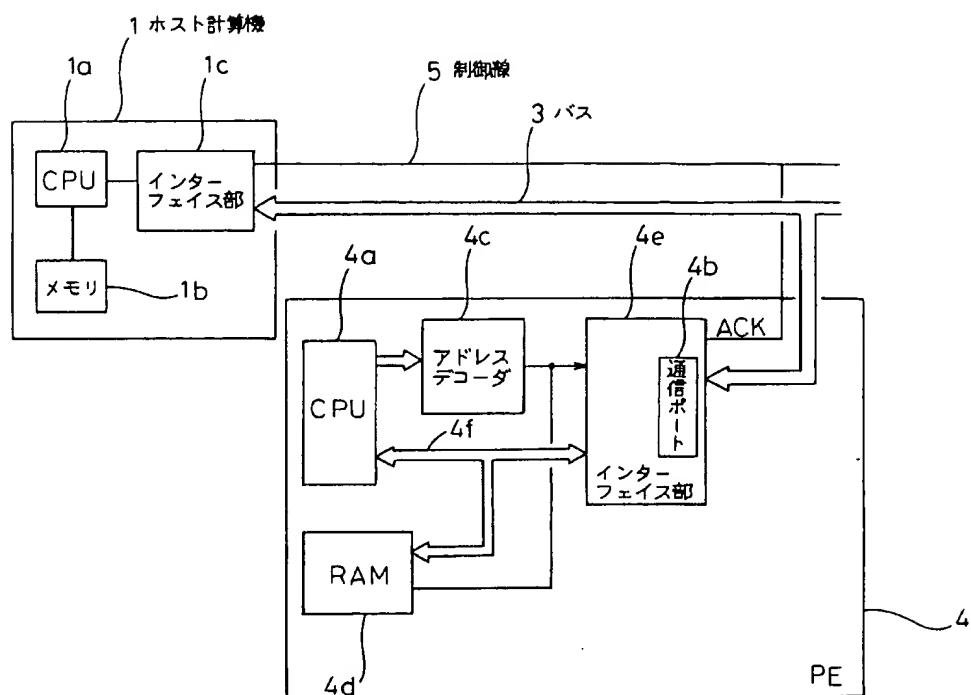
ホスト計算機がPEに送るデータとPEのCPUが実行する命令例
(従来) を示す図

第3 図

ホスト計算機がPEに送るデータ	PEのCPUが実行する命令
t1 時間 LD F000 OS0	LD F000 ; OS0がレジスタに入る ST 2000 ; OS0を2000番地に書く
ST 2000	LD F000 ; OS1がレジスタに入る ST 2001 ; OS1を2001番地に書く
LD F000 OS1	LD F000 ; OS2がレジスタに入る ST 2002 ; OS2を2002番地に書く
ST 2001	LD F000 ; OS3がレジスタに入る ST 2003 ; OS3を2003番地に書く
LD F000 OS2	⋮
ST 2002	⋮
LD F000 OS3	⋮
ST 2003	⋮
LD F000 OS4	⋮
⋮	⋮
LD F000 OSZ	⋮
ST 20FF	⋮ ; OSのスタートアドレスから実行を開始

ホスト計算機がPEに送るデータとPEのCPUが実行する命令例
(本発明) を示す図

第4 図



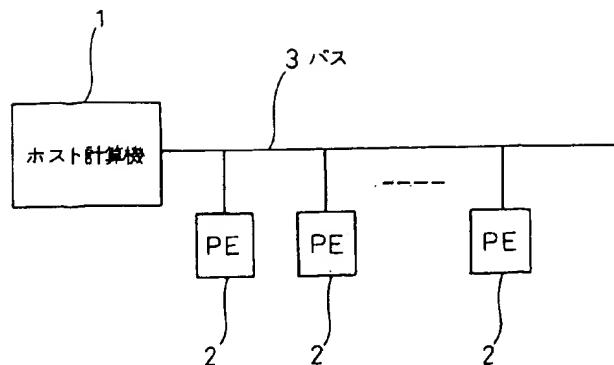
本発明の一実施例を示す構成ブロック図

第 5 図

PE側	ホスト計算機側
(1) リセット	① PEをリセット
(2) 命令をアドレス0からフェッチ (通信ポートをアクセス)	
(3) ACKが有効になるまでホールド	② 全PEが通信ポートをアクセスするのを待って、PEの第1命令を通信ポートに出力する
(4) 第1命令を読み込み実行	③ ACKが有効になる
(5) 次の命令を次アドレスからフェッチ (通信ポートをアクセス)	
(6) ACKが有効になるまでホールド	④ 全PEが通信ポートをアクセスするのを待って、PEの次命令を通信ポートに出力する
(7) 命令を読み込み実行	⑤ ACKが有効になる
(8) 命令アドレスが1FFFを越えない間、 (5), (6), (7)を繰り返す	⑥ ブートシーケンスが終了するまで、 ④, ⑤を繰り返す

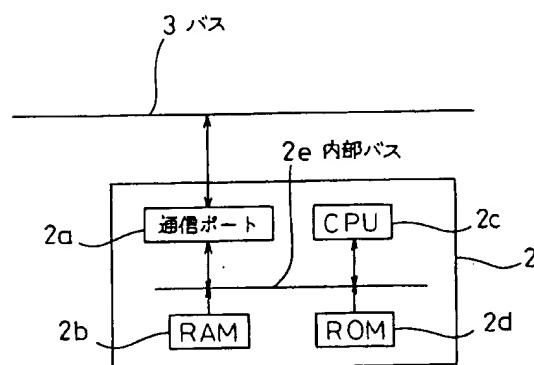
本発明によるブートシーケンスを示す図

第6回



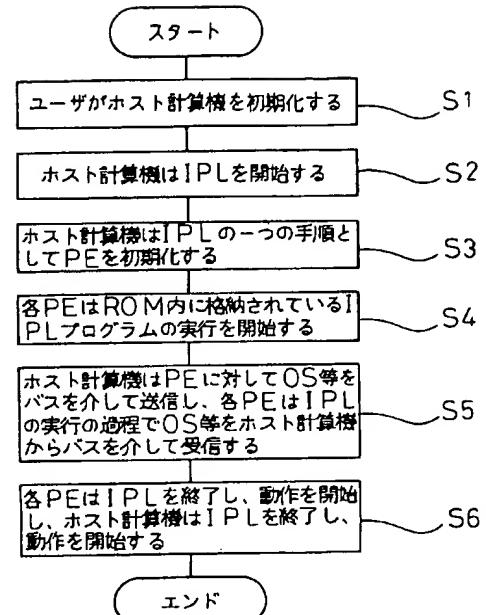
従来の分散メモリ型並列計算機の構成ブロック図

第 7 図



各PEの内部構成例(従来を示す図)

第 8 図



従来のブートアップの手順を示すフローチャート

第 9 図